

03/22/99  
jc376 U.S. PTO  
09274294-032299

7  
jc135 U.S. PTO  
09/274294  
03/22/99

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship ..... Gunter et al.  
Applicant ..... Microsoft Corporation  
Attorney's Docket No. .... MS1-298US  
Title: System and Method for Trusted Inspection of a Data Stream

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks  
Washington, D.C. 20231  
From: Lewis C. Lee (509) 324-9256  
Lee & Hayes, PLLC  
W. 201 North River Drive, Suite 430  
Spokane, WA 99201

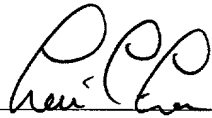
The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Transmittal Letter with Certificate of Mailing included.
2. PTO Return Postcard Receipt
3. New patent application (title page plus 25 pages, including claims 1-20 & Abstract)
4. Executed Declaration
5. 4 sheets of formal drawings (Figs. 1-4)
6. Assignment w/Recordation Cover Sheet

Large Entity Status ☒ Small Entity Status ☐

The Commissioner is hereby authorized to charge payment of fees or credit overpayments to Deposit Account No. 50-0463 in connection with any patent application filing fees under 37 CFR 1.16, and any processing fees under 37 CFR 1.17.

Date: March 22, 1999

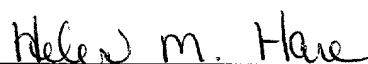
By:   
Lewis C. Lee  
Reg. No. 34,656

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL209423280

Date: March 22, 1999

By:   
Helen M. Hare

652250 4524260

CALCULATION OF TOTAL FEES DUE				MSI-298US		
CLAIMS FEES	Number Filed (Col. 1)	No. Extra (Col. 2)	Small Entity		Large Entity	
			Rate (\$)	Fee (\$)	Rate (\$)	Fee (\$)
Basic Fee				395		760
Total Claims	20- 20 =		x 9 =		x 18 =	
Indep. Claims	7- 3 =		x 39 =		4x 78 =	312
						1,072
TOTAL APPLICATION FEES						
[						
Any Other Fees						
TOTAL FEES						1,072

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**System and Method for Trusted Inspection of a Data  
Stream**

Inventor(s):  
David Gunter  
Leeon Moshe Shachaf

ATTORNEY'S DOCKET NO. MS1-298US

09423280 "4634260

1 **TECHNICAL FIELD**

2 This invention relates to network systems and protected communication  
3 over network systems. More particularly, this invention relates to systems and  
4 methods for inspecting protected data communication in a trusted manner.  
5

6 **BACKGROUND**

7 The Internet and World Wide Web ("Web") have grown rapidly into an  
8 essential backbone for business communications. The Internet, however, is a  
9 public and open network that offers no inherent protection to data communication.  
10 Yet, today's business environment (as well as others) demand manageable and  
11 secure data communication over such public networks.

12 Many organizations are now using the Internet as a data transmission  
13 medium between different proprietary networks. To enable this communication  
14 over an otherwise public network, data is commonly encrypted into a protected  
15 form. This affords essentially the same protection and security as a private  
16 network, while benefiting from the flexibility and lower costs offered by the  
17 Internet. To this end, the concept of a virtual private network (VPN) has been  
18 developed.

19 A virtual private network provides a secure, authenticated mechanism for  
20 communicating between two endpoints, such as between two networks. The VPN  
21 establishes an encrypted data flow between the two endpoints. Since there is not  
22 an actual private network connection in place, and the data is actually being routed  
23 over the Internet, this data flow can be thought of as a "tunnel" through the  
24 Internet. Data conceptually enters the tunnel at one end and emerges, secure and  
25 unchanged, at the other end.

To ensure that the transmission is secure, the data must be protected from unauthorized access during transmission over the Internet. Consider how a malicious party could access this data. A malicious party could read and record the data, or modify it in some way, or even replace the valid data with different data. The source of the data could be disguised by changing the electronic source address of the Internet Protocol (IP) header in the data stream.

Data transmitted via a VPN is therefore encrypted to prevent any inspection or modification. Various protocols exist that provide security and authentication features for VPNs. Point-to-Point Tunneling Protocol (PPTP) and the Internet Engineering Task Force's IPSEC specification are the two most common VPN protocols.

Point-to-point tunneling protocol (PPTP) is a network protocol that enables the secure transfer of data from a remote client to a private enterprise server by creating a virtual private network across TCP/IP-based data networks. PPTP supports on-demand, multi-protocol, and virtual private networking over public networks such as the Internet. The networking technology of PPTP is an extension of the remote access point-to-point protocol defined in the document by the Internet Engineering Task Force (IETF) titled "The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams over Point-to-Point Links," referred to as RFC 1171. PPTP is a network protocol that encapsulates PPP packets into IP datagrams for transmission over the Internet or other public TCP/IP-based networks. PPTP can also be used in private LAN-to-LAN networking.

IP Security (IPSEC) is a network protocol designed by the Internet Engineering Task Force (IETF) for the Internet Protocol that supports network-

level authentication, data integrity, and encryption. Encryption is encoded via a shared secret key, typically generated by the Diffie-Hellman Key Agreement algorithm. IP Security uses an Authentication Header (AH) and an Encapsulated Security Payload (ESP). The authentication header provides data communication with source authentication and integrity. The encapsulated security payload provides confidentiality in addition to authentication and integrity. With IP security, only the sender and recipient know the security key. If the authentication data is valid, the recipient knows that the communication came from the sender, and that it was not changed in transit.

The data payload in packets transmitted over a VPN is encrypted. The routing information, such as the source and destination addresses, is left unencrypted to enable routing through the network. However, it is possible to secure the routing components by calculating a value, such as a checksum, based on the contents of the address fields and then encrypt the checksum along with the data payload.

Fig. 1 shows one example of encrypting a data packet according to IPSEC. A data packet 20 comprises a data payload 22 and routing header 24 that contains addressing information. A hashing unit 26 cryptographically hashes the entire packet, including the routing header, using a hashing algorithm (e.g., MD4, MD5) and a resultant hash value (HV) 28 is appended to the packet. An encrypting unit 30 then encrypts the data payload 22 and appended hash value 28. The unencrypted routing header is appended to the encrypted portion 32 and the packet is sent out.

The destination receives the packets and decrypts the encrypted portion 32. It then recalculates the hash value from the decrypted data payload and the routing

header. If a malicious user were to attempt to rewrite any of the unencrypted data, the hash value calculated by the receiver would not match the accompanying hash value 28, and the unauthorized modification would be detected.

A drawback in these VPN protocols is that the encryption mechanisms prevent all data stream inspection, even inspection for legitimate purposes. There are situations when a party would like to access and inspect the encrypted data before it reaches its final destination. For example, a network access point such as a firewall or proxy server may want to perform virus scanning or implement policy based access controls. The firewall might like to strip off one address (say a general address to a company that is given out publicly) and replace it with internal addresses used behind the firewall. However, since the original address is used to create the encrypted payload, it cannot be later stripped away and replaced with another address without detrimentally hindering the destination's ability to restore the original packet. The only way to engage in legitimate inspection is to have access to the unencrypted data stream, which is counter to the basic design of a virtual private network.

Accordingly, there is a need for a secure mechanism that allows a trusted computer system to function as a trusted man-in-the-middle and have access to the unencrypted version of the data flowing through a virtual private network.

## **SUMMARY**

The invention concerns a network architecture in which two endpoints communicate via a virtual private network (VPN) on an otherwise public network, such as the Internet, and an intermediary is permitted to inspect the data communication in a secure and trusted manner.

1 In one implementation, the network architecture has an external client and  
2 an internal client that exchange encrypted data over a network. The internal client  
3 is coupled to the network via a network access point, such as a firewall/proxy  
4 server. All three participants have their own pair of public/private keys. An  
5 independent key server holds the public keys for all three participants.

6 The external and internal clients establish a virtual private network by  
7 negotiating a session key used to encrypt data being exchanged between them.  
8 Initially, only the clients know the session key, and not the firewall. To grant the  
9 firewall trusted access to the data stream on the VPN, the internal client securely  
10 transfers the session key to the firewall. The internal client requests and receives  
11 the firewall's public key from the key server and encrypts the session key using  
12 the firewall's public key. The internal client then signs the encrypted key by  
13 encrypting it using the internal client's private key.

14 The firewall authenticates the signature by decrypting the message using  
15 the internal client's public key (obtained from the key server or directly from the  
16 internal computer). The firewall then decrypts the session key using its own  
17 private key. If the dual decryption yields a valid key, the firewall is assured that  
18 the session key was sent by the internal client and was not subsequently altered or  
19 tampered with in route.

20 Once the session key is transferred, the firewall is able to decrypt the data  
21 stream on the VPN. The firewall can now unintrusively inspect the data stream in  
22 a manner that is transparent to the external and internal clients.  
23  
24  
25



1 **BRIEF DESCRIPTION OF THE DRAWINGS**

2 Fig. 1 shows a prior art packet encryption process employed by a  
3 conventional virtual private network protocol.

4 Fig. 2 shows a network system having external and internal computing  
5 units that communicate securely over a network, and a trusted intermediary  
6 computing unit that has trusted access to the encrypted data stream.

7 Fig. 3 shows an exemplary computer that can be configured as one of the  
8 computing units in the network system.

9 Fig. 4 is a flow diagram showing steps in a method implemented for trusted  
10 inspection of a data stream.

11  
12 **DETAILED DESCRIPTION**

13 This invention concerns a network architecture that enables secure  
14 communication between two endpoints via a virtual private network (VPN) on an  
15 otherwise public network, such as the Internet. The architecture allows an  
16 intermediary to gain trusted access to the data stream in the VPN. Trusted access  
17 is achieved through a combination of public and private key protocols. Toward  
18 that end, the following discussion assumes that the reader is familiar with  
19 cryptography. For a basic introduction of cryptography, the reader is directed to a  
20 text written by Bruce Schneier and entitled "Applied Cryptography: Protocols,  
21 Algorithms, and Source Code in C," published by John Wiley & Sons with  
22 copyright 1994 (or second edition with copyright 1996).

## Network Architecture

Fig. 2 shows a network system 40 having an external client computer 42 that communicates with an internal client computer 44 via a public network, such as the Internet 46. The two computers employ a virtual private network (VPN) through the Internet 46 to communicate with one another. Data is communicated through a network access point, embodied in this example as a firewall computer 48, which resides on the internal client's side of the Internet 46. A proprietary intranet 50 connects the firewall 48 and the internal client 44. The firewall 48 is configured with proxy server software 52, such as Microsoft Proxy Server, to support firewall functionality.

All three participants—external client 42, internal client 44, and firewall 48—are equipped with cryptography modules 54, 56, and 58, respectively. Each cryptography module enables various cryptographic functionality, including encryption, decryption, hashing, authentication, and signing. All three participants also have their own pair of public/private keys. The private keys are stored at the respective computers, as represented by external client private key ( $EC_{pri}$ ) 60 at external client 42, internal client private key ( $IC_{pri}$ ) 62 at internal client 44, and firewall computer private key ( $FC_{pri}$ ) 64 at firewall 48.

An independent key server 66 holds corresponding public keys 68, 70, and 72 for external client 42, internal client 44, and firewall 48, respectively.

All three cryptography modules 54, 56, and 58 contain similar components. At external client 42, the cryptography module 54 has a session key generator 74 that generates secret session keys used in communication over the VPN and a hashing unit 76 that employ hashing algorithms (e.g., MD4, MD5, etc.) to compute hash digests for use in encrypting the data packets or for purposes of

creating a digital signature. The external client 42 also has symmetric encryption/decryption unit 78 that encrypts or decrypts items using a symmetric key algorithm, and an asymmetric encrypt/decryption unit 80 that encrypts or decrypts items using an asymmetric key algorithm.

At internal client 44, the cryptography module 56 has a session key generator 82, a hashing unit 84, a symmetric encrypt/decrypt unit 86, and an asymmetric encrypt/decrypt unit 88. At firewall 48, the cryptography module 58 has a hashing unit 90, a symmetric encrypt/decrypt unit 92, and an asymmetric encrypt/decrypt unit 94.

The “asymmetric” algorithm employed in the asymmetric encrypt/decrypt units 80, 88, and 94 uses the public and private key pairs. The separate keys are based upon a mathematical relationship in which one key cannot be calculated from the other key. Encryption and decryption using an asymmetric key pair can be represented as follows:

$$\text{Asym.E}_{K_{\text{pri}}} (M) = M.\text{enc}$$

$$\text{Asym.D}_{K_{\text{pub}}} (M.\text{enc}) = M$$

where “Asym.E<sub>K<sub>pri</sub></sub>” is an encryption function using a private key “K<sub>pri</sub>,” “M” is a message, “M.enc” is an encrypted version of the plaintext message, and “Asym.D<sub>K<sub>pub</sub></sub>” is a decryption function using the public key “K<sub>pub</sub>”. The inverse is also true in that a message can be encrypted using the public key and then decrypted using the private key. In a public key system, the public key can be distributed to other parties, such as the key server 66, and the private key is maintained in confidence at the respective party. An example asymmetric

1 algorithm is the well-known RSA cryptographic algorithm named for the creators  
2 Rivest, Shamir, and Adleman.

3 The cryptography modules 54, 56, and 58 may be implemented in many  
4 ways. In one implementation, the module is embodied as a cryptographic API  
5 (Application Program Interface) exposed in Windows-brand operating systems,  
6 such as the Windows NT operating system. The cryptographic API ("CAPI" or  
7 "CryptoAPI") is described in U.S. Patent No. 5,689,565, entitled "Cryptography  
8 System and Method for Providing Cryptographic Services for a Computer  
9 Application", which is assigned to Microsoft Corporation. This patent is hereby  
10 incorporated by reference.

11 According to the network architecture in Fig. 2, the external and internal  
12 clients 42, 44 establish a virtual private network connection through the Internet  
13 46. The computers engage in a key negotiation process to negotiate a session key.  
14 The key negotiation process is dependent upon the VPN communication protocol.  
15 In the illustrated example, the participants select a shared symmetric session key.  
16 Once a shared session key is found, the external and internal clients begin an  
17 encrypted communication session.

18 Firewall 48 is permitted to inspect the data stream passing through it on the  
19 VPN. The internal client 44 transmits the shared session key to the firewall 48 in  
20 a secure manner, preferably via public key technology. Once the firewall has the  
21 session key, it can dynamically decrypt traffic in the VPN data stream and monitor  
22 its content. Trusted inspection of the data stream is completely transparent to both  
23 parties in the VPN communication.

24 One exemplary implementation of this process is described below under the  
25 heading "Operation" and with reference to Fig. 4. Prior to explaining this process,

1 however, an exemplary implementation of a computer used to implement anyone  
2 of the participants is described.

### 4 **Exemplary Computer**

5 Fig. 3 shows an exemplary implementation of a computer, such as the  
6 external client 42, the internal client 44, firewall 58, or key server 66. The host  
7 computer is a general-purpose computing device in the form of a conventional  
8 personal computer 100 that is configured to operate as a network server (in the  
9 case of the firewall and key server computers) or as a client computer (in the case  
10 of the external and internal clients).

11 Computer 100 includes a processing unit 102, a system memory 104, and a  
12 system bus 106 that couples various system components including the system  
13 memory 104 to the processing unit 102. The system bus 106 may be any of  
14 several types of bus structures including a memory bus or memory controller, a  
15 peripheral bus, and a local bus using any of a variety of bus architectures. The  
16 system memory 104 includes read only memory (ROM) 108 and random access  
17 memory (RAM) 110. A basic input/output system 112 (BIOS) is stored in ROM  
18 108.

19 Computer 100 also has one or more of the following drives: a hard disk  
20 drive 114 for reading from and writing to a hard disk, a magnetic disk drive 116  
21 for reading from or writing to a removable magnetic disk 118, and an optical disk  
22 drive 120 for reading from or writing to a removable optical disk 122 such as a CD  
23 ROM or other optical media. The hard disk drive 114, magnetic disk drive 116,  
24 and optical disk drive 120 are connected to the system bus 106 by a hard disk  
25 drive interface 124, a magnetic disk drive interface 126, and an optical drive

1 interface 128, respectively. The drives and their associated computer-readable  
2 media provide nonvolatile storage of computer readable instructions, data  
3 structures, program modules and other data for the personal computer. Although a  
4 hard disk, a removable magnetic disk and a removable optical disk are described,  
5 other types of computer readable media can be used to store data, such as flash  
6 memory cards, digital video disks, random access memories (RAMs), read only  
7 memories (ROM), and the like.

8 A number of program modules may be stored on the hard disk, magnetic  
9 disk, optical disk, ROM, or RAM. These programs include an operating system  
10 130, one or more application programs 132, other program modules 134, and  
11 program data 136. The programs 132 or modules 134, for example, include the  
12 cryptography module installed at each participant.

13 A user may enter commands and information into the personal computer  
14 100 through input devices such as keyboard 138 and pointing device 140. Other  
15 input devices (not shown) may include a microphone, joystick, game pad, satellite  
16 dish, scanner, or the like. These and other input devices are often connected to the  
17 processing unit 102 through a serial port interface 142 that is coupled to the  
18 system bus 106, but may be connected by other interfaces, such as a parallel port,  
19 game port, or a universal serial bus (USB). A monitor 144 or other type of display  
20 device is also connected to the system bus 106 via an interface, such as a video  
21 adapter 146. In addition to the monitor, personal computers typically include other  
22 peripheral output devices (not shown) such as speakers and printers.

23 The server computer 100 is connected to a network 148 (e.g., Internet 46,  
24 intranet 50) through a network interface or adapter 150, a modem 152, or other  
25 means for establishing communications over the network. The modem 152, which

may be internal or external, is connected to the system bus 106 via the serial port interface 142.

### **Operation**

The network architecture enables trusted inspection of a data stream on a virtual private network established between the external and internal clients. As background, it is assumed that the participating computers have previously generated their own pairs of private and public keys. It is further assumed that the private keys are installed locally in a registry at each respective computer and the public keys are stored in a file system or system registry at the key server 66.

Fig. 4 shows steps in the trusted inspection process. The steps are performed in software, hardware, or a combination of hardware and software. The steps are described with respect to the architecture of Fig. 2.

At steps 200 and 202, the external client 42 and internal client 44 establish a virtual private network connection by negotiating a session key (SK). The firewall 48 opens appropriate ports to allow the VPN key negotiation process to proceed. The key negotiation process is specific to the VPN protocol. In most key negotiation processes, the two endpoint systems use a combination of public and private keys to generate and exchange a session key. In some cases, such as with PPTP, a session key is derived using a binary challenge. The key negotiation process is designed to prevent cleartext transmission of the session key. The session key is either (1) generated and transmitted via public key encryption or (2) derived independently by both parties, as in the case of the well-known Diffie-Hellman key exchange. The external and internal clients employ session key

generators 74 and 82, respectively, to define the session key and their own key pairs to exchange the session key securely.

Once the external and internal clients 42, 44 have selected and communicated a shared session key (SK), they are able to tunnel encrypted data through the Internet. To enable trusted data stream inspection, the firewall 48 needs to know the shared session key. The VPN participant that operates on the same intranet 50 with the firewall 48 (in this case, the internal client 44) is responsible for getting the shared session key to the firewall 48 in a secure manner.

At step 204, the internal client 44 requests and receives the firewall's public key  $FC_{pub}$  70 from the key server 66. Alternatively, the firewall passes its public key directly to the internal client 44 over the intranet 50. The internal client 44 employs the asymmetric encryption/decryption unit 88 to encrypt the shared session key SK using the firewall's public key  $FC_{pub}$  70 (step 206). This is represented as follows:

$$Asym.E_{FC_{pub}}(SK) = SK.enc$$

where " $Asym.E_{FC_{pub}}$ " is an asymmetric encryption function using the firewall computer's public key " $FC_{pub}$ ", and the resultant " $SK.enc$ " is an encrypted version of the session key. Since only the firewall 48 knows the corresponding private key  $FC_{pri}$ , only the firewall will be able to decrypt the session key, thereby ensuring secure transfer.



At step 208, the internal client digitally signs the encrypted session key "SK.enc" by encrypting a message containing the encrypted session key, or a hash digest of it, using its own private key IC<sub>pri</sub> 62. This is represented as follows:

$$\text{Asym.E}_{\text{ICpri}}(\text{SK.enc}) = \text{Signed}(\text{SK.enc})_{\text{IC}}$$

where "Asym.E<sub>ICpri</sub>" is an asymmetric encryption function using the internal client's private key "IC<sub>pri</sub>", and the output "Signed(SK.enc)<sub>IC</sub>" is the signed result. With the signature, the firewall can authenticate that the message came from the internal client and not an imposter.

At step 210, the internal client 44 transfers the signed, encrypted session key over the intranet 50 to the firewall 48.

When the firewall 48 receives the signed encrypted session key, it first authenticates the signature by decrypting the message using the internal client's public key IC<sub>pub</sub> 72 (step 212). The firewall retrieves this public key from the key server 66, or receives it in the message from the internal client 44. Alternatively, the internal client's public key is already pre-cached at the firewall. The firewall invokes the asymmetric encryption/decryption unit 94 to perform the following decryption process:

$$\text{Asym.D}_{\text{ICpub}}(\text{Signed}(\text{SK.enc})_{\text{IC}}) = \text{SK.enc}$$

where "Asym.D<sub>ICpub</sub>" is an asymmetric decryption function using the internal client's public key "IC<sub>pub</sub>". At step 214, the firewall uses the asymmetric

1 encryption/decryption unit 94 to decrypt the session key SK using its own private  
2 key "FC<sub>pri</sub>", as follows:

$$3 \quad 4 \quad \text{Asym.D}_{\text{FCpri}}(\text{SK.enc}) = \text{SK}$$

5  
6 where "Asym.D<sub>FCpri</sub>" is an asymmetric decryption function using the firewall  
7 computer's private key "FC<sub>pri</sub>". At this point, the session key has been securely  
8 transferred to the firewall.

9 Once the firewall 48 gains possession of the session key, it can dynamically  
10 decrypt traffic in the VPN data stream between the external and internal clients,  
11 and monitor the content of the data stream. Trusted data stream is completely  
12 transparent to both parties in the VPN communication. It is noted that the trusted  
13 inspection process is applicable for VPN protocols that require exchanges to take  
14 place at regular intervals. Whenever a new key negotiation session completes, the  
15 internal client simply forwards the new shared session key to the network access  
16 point as it did initially.

17 At step 216, the external client 42 calls the symmetric  
18 encryption/decryption unit 78 to encrypt data destined for the internal client 44  
19 using the session key SK negotiated at steps 200 and 202. This is represented as  
20 follows:

$$21 \quad 22 \quad \text{Sym.E}_{\text{SK}}(\text{Data}) = \text{Data.enc}$$

23  
24 where "Sym.E<sub>SK</sub>" is a symmetric encryption function using a session key "SK" and  
25 "Data.enc" is an encrypted version of the data. The encrypted data is transmitted

over the Internet 46 (step 218), where it is intercepted at the firewall 48 (step 220). The firewall 48 invokes symmetric encryption/decryption unit 92 to decrypt the encrypted data stream using the same session key SK that it received from the internal client 44 (step 222), as follows:

$$\text{Sym.D}_{\text{SK}}(\text{Data.enc}) = \text{Data}$$

where "Sym.D<sub>SK</sub>" is a symmetric decryption function using a session key "SK". Once the data is decrypted, the firewall 48 has several options available. It can forward the encrypted data onto the internal client (step 224). It can perform tasks based on the unencrypted data, such making internal routing decisions or other policy considerations. The firewall 48 may also cache or store a decrypted version of the data while forwarding on the encrypted version to the internal client.

### **Conclusion**

The invention advantageously provides a network architecture that allows legitimate trusted inspection of a VPN data stream at an intermediary, such as a firewall or proxy server. Public key encryption and signing techniques are employed to securely transfer a VPN session key from a VPN endpoint to the firewall. The cryptography techniques allow the firewall to authenticate that the session key came from the VPN endpoint and was not subsequently tampered with by a third party.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or

1 steps described. Rather, the specific features and steps are disclosed as preferred  
 2 forms of implementing the claimed invention.  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25

1 **CLAIMS**

2 1. A method for inspecting an encrypted data stream being transferred  
3 over a network between two endpoints, the data stream being encrypted using a  
4 session key known to both endpoints, the method comprising:

5 securely transferring the session key from one of the endpoints to an  
6 intermediary having access to the encrypted data stream;

7 decrypting the encrypted data stream at the intermediary using the session  
8 key; and

9 inspecting the data stream following decryption.

10  
11 2. A method as recited in claim 1, wherein securely transferring  
12 comprises:

13 encrypting the session key using a public key associated with the  
14 intermediary; and

15 sending the encrypted session key to the intermediary.

16  
17 3. A method as recited in claim 1, wherein securely transferring  
18 comprises:

19 encrypting the session key using a public key associated with the  
20 intermediary;

21 signing the encrypted session key using a private key associated with the  
22 intermediary; and

23 sending the signed and encrypted session key to the intermediary.  
24  
25

1           4.    A method as recited in claim 1, further comprising storing the data  
2 stream at the intermediary.

3  
4           5.    A method for inspecting an encrypted data stream being transferred  
5 over a network between two endpoints and via an intermediary, the data stream  
6 being encrypted using a session key known to both endpoints, the method  
7 comprising:

8               storing a public key from a public/private key pair associated with one of  
9 the endpoints at a key storage;

10              storing a public key from a public/private key pair associated with the  
11 intermediary at the key storage;

12              obtaining, at said one endpoint, the intermediary's public key from the key  
13 storage;

14              encrypting, at said one endpoint, the session key using the intermediary's  
15 public key to produce an encrypted session key;

16              encrypting, at said one endpoint, the encrypted session key using a private  
17 key from the public private key pair associated with said one endpoint to produce  
18 a signed encrypted session key;

19              passing the signed encrypted session key to the intermediary;

20              obtaining, at the intermediary, the one endpoint's public key from the key  
21 storage;

22              decrypting, at the intermediary, the signed encrypted session key using the  
23 one endpoint's public key to return the encrypted session key;

24              decrypting, at the intermediary, the encrypted session key using the  
25 intermediary's private key to return the session key; and

1 using the session key at the intermediary to decrypt the encrypted data  
2 stream.

3  
4 6. In a network system in which an encrypted data stream is transferred  
5 over a network between two endpoints and via an intermediary, the data stream  
6 being encrypted using a session key known to both endpoints, computer-readable  
7 media at one of the endpoints and at the intermediary storing computer-executable  
8 instructions for performing the method as recited in claim 5.

9  
10 7. In a network system having an external client that exchanges  
11 encrypted data with an external client over a network and through a firewall  
12 intermediate of the internal and external clients, the encrypted data being  
13 encrypted using a session key known to the internal and external clients, a method  
14 executed at the firewall comprising:

15 receiving an encrypted and signed session key from the internal client, the  
16 encrypted and signed session key bearing a digital signature of the internal client;

17 authenticating the digital signature as belonging to the internal client;

18 decrypting the session key; and

19 decrypting the encrypted data being exchanged between the internal and  
20 external clients using the session key.

1           8.     A method as recited in claim 7, wherein the encrypted and signed  
2 session key is encrypted using a public key from a public/private key pair  
3 associated with the firewall, and the decrypting comprises decrypting the session  
4 key using a private key from the public/private key pair.

5  
6           9.     A method as recited in claim 7, further comprising inspecting the data  
7 in an unencrypted form.

8  
9           10.    A method as recited in claim 7, further comprising storing the data  
10 in an unencrypted form.

11  
12           11.    In a network system having an external client that exchanges  
13 encrypted data with an external client over a network and through a firewall  
14 intermediate of the internal and external clients, the encrypted data being  
15 encrypted using a session key known to the internal and external clients, a  
16 computer-readable medium resident at the firewall storing computer-executable  
17 instructions for performing method as recited in claim 7.

18  
19           12.    A network system comprising:  
20           an internal client and an external client configured to communicate  
21 encrypted data over a network using virtual private network communication, the  
22 data being encrypted using a session key;  
23           an intermediary having access to the encrypted data being communicated  
24 between the internal client and the external client;





key to decrypt the encrypted data as it is being exchanged between the two endpoints.

17. A software architecture as recited in claim 16, wherein intermediary-resident code inspects the data in unencrypted form.

18. A software architecture as recited in claim 16, wherein intermediary-resident code stores the data in unencrypted form.

19. In a network system having an external client that exchanges encrypted data with an external client over a network and through a firewall intermediate of the internal and external clients, the encrypted data being encrypted using a session key known to the internal and external clients, computer-readable media distributed at the internal client and the firewall storing computer-executable instructions for:

encrypting the session key at the internal client;

signing the encrypted session key with a digital signature associated with the internal client;

passing the signed and encrypted session key to the intermediary;

authenticating, at the intermediary, the digital signature of the internal client;

decrypting the session key at the intermediary;

decrypting, at the intermediary, the encrypted data using the session key;

and

inspecting the data in route between the internal and external clients.

1  
2       **20.**     In a network system in which an encrypted data stream is transferred  
3 over a network between two endpoints and via an intermediary, the data stream  
4 being encrypted using a session key known to both endpoints, computer-readable  
5 media at one of the endpoints and at the intermediary storing computer-executable  
6 instructions for:

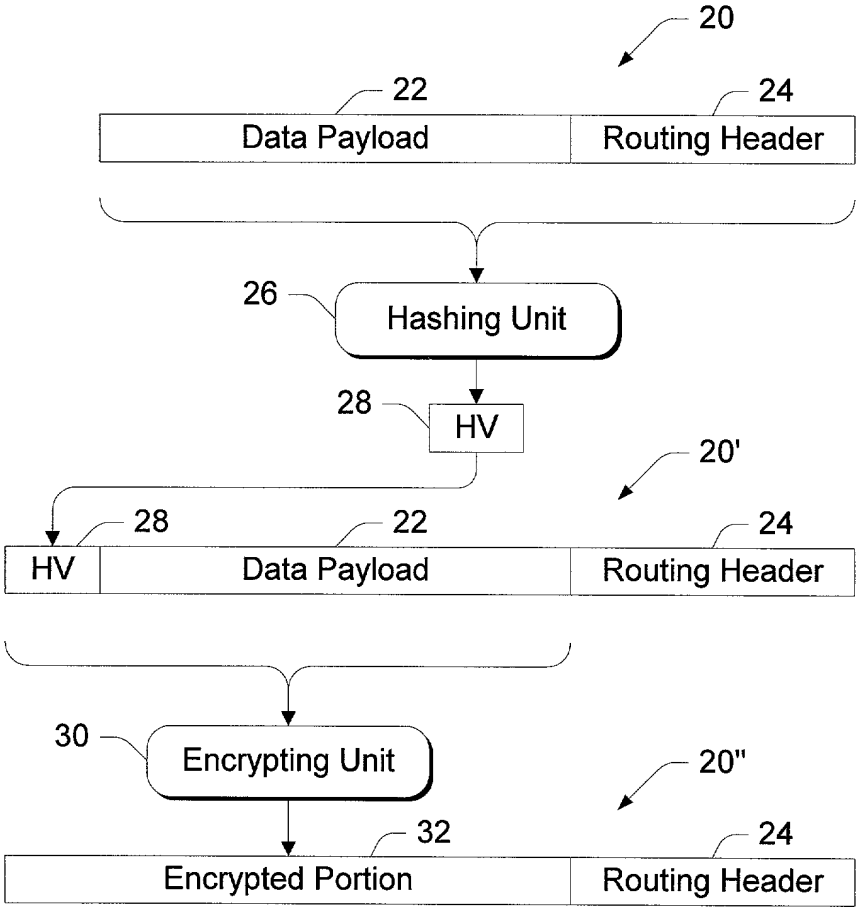
7             securely transferring the session key from one of the endpoints to an  
8 intermediary having access to the encrypted data stream;

9             decrypting the encrypted data stream at the intermediary using the session  
10 key; and

11            inspecting the data stream following decryption.  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

# **ABSTRACT**

A network architecture allows an intermediary to inspect an encrypted data stream on a virtual private network (VPN) in a secure and trusted manner. The endpoints establish a virtual private network by negotiating a session key used to encrypt data being exchanged between them. The endpoints know the session key, but not the intermediary. To grant the intermediary trusted access to the data stream on the VPN, one endpoint securely transfers the session key to the firewall by encrypting the session key using the intermediary's public key and then signing the encrypted session key. The intermediary authenticates the signature and decrypts the session key using its own private key. If the process yields a valid key, the intermediary is assured that the session key was sent by the endpoint and was not subsequently tampered with in route. Once the session key is transferred, the firewall can decrypt and inspect the data stream on the VPN in a manner that is transparent to the endpoints.



*Fig. 1*  
*Prior Art*

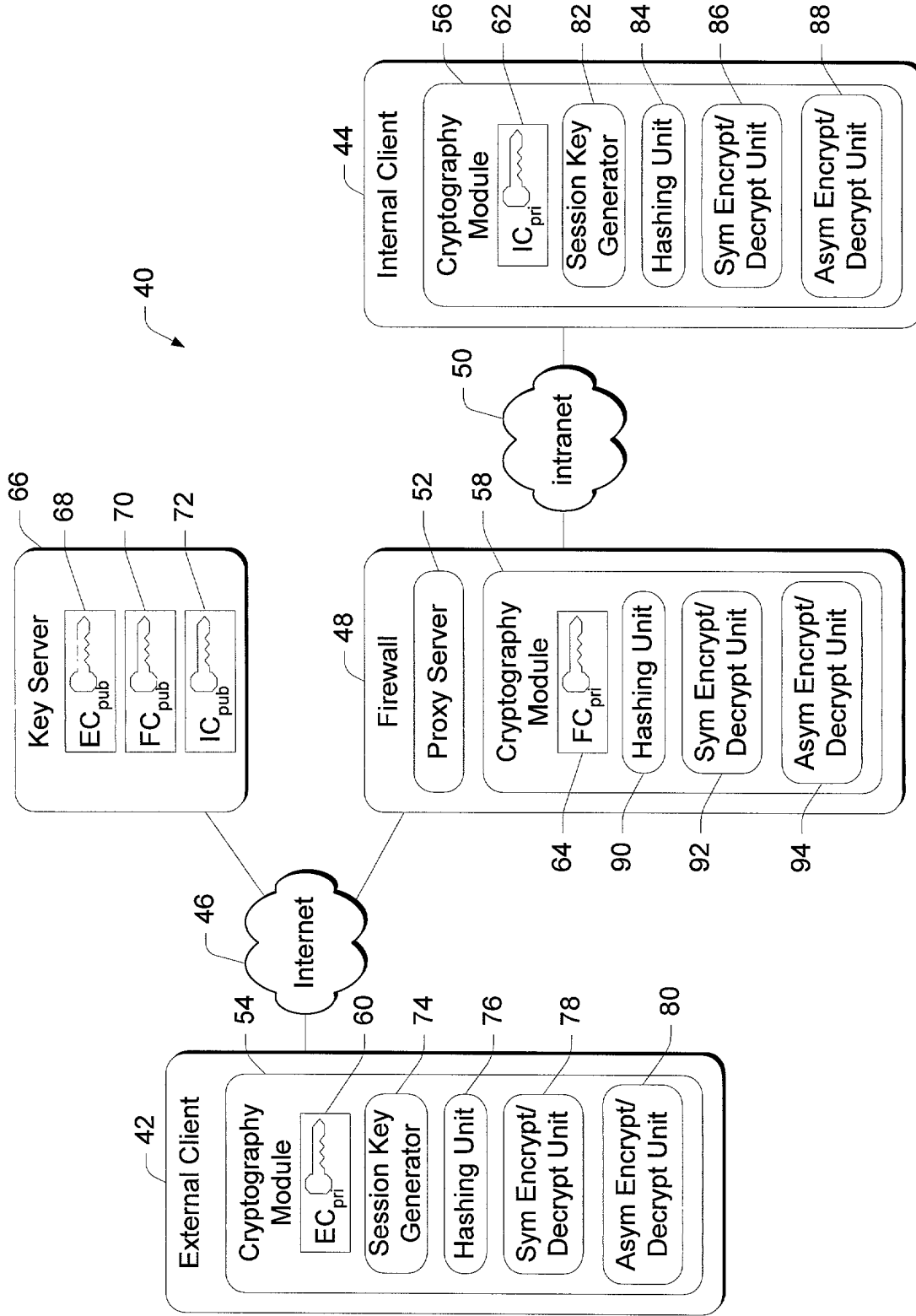
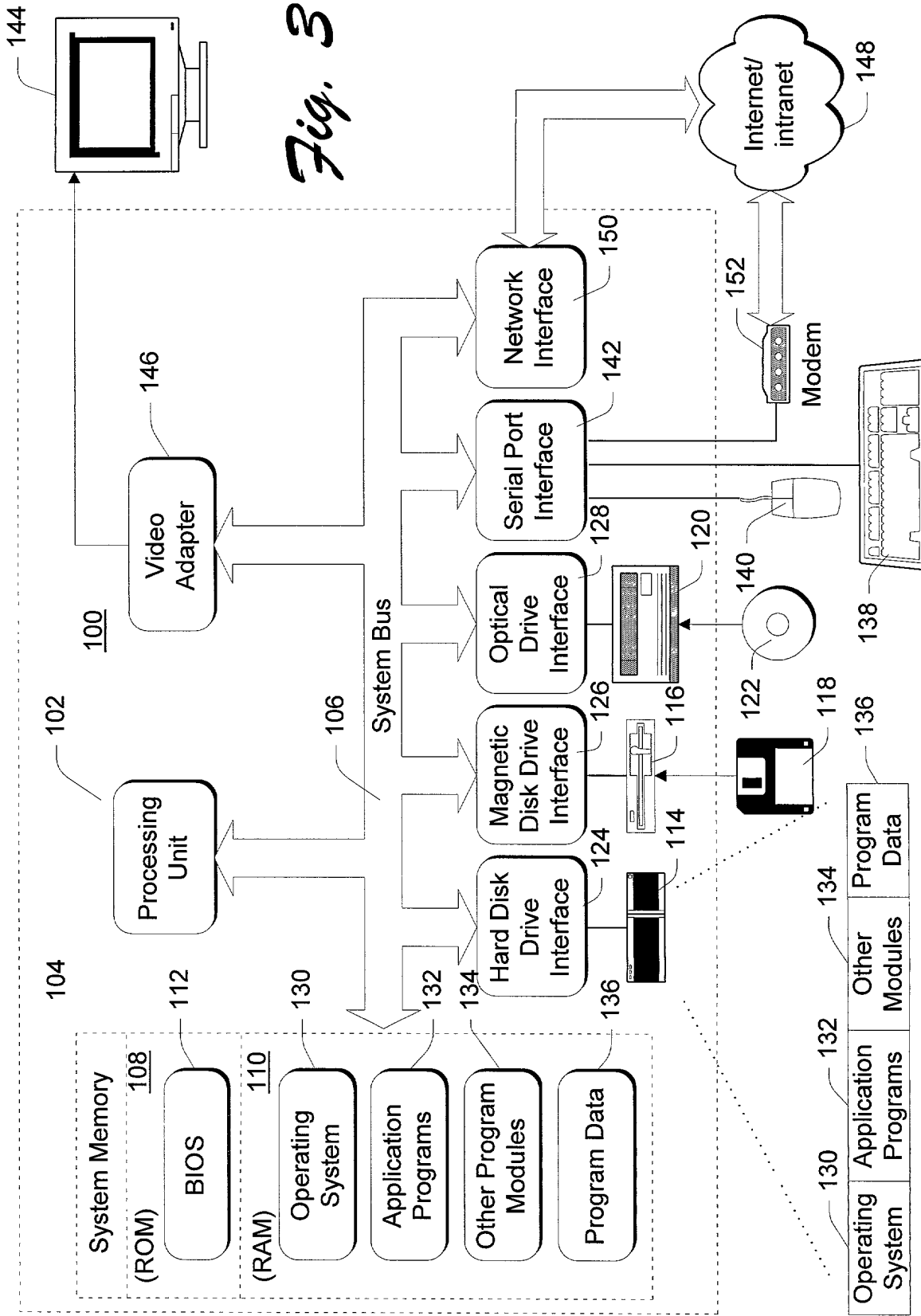


Fig. 2



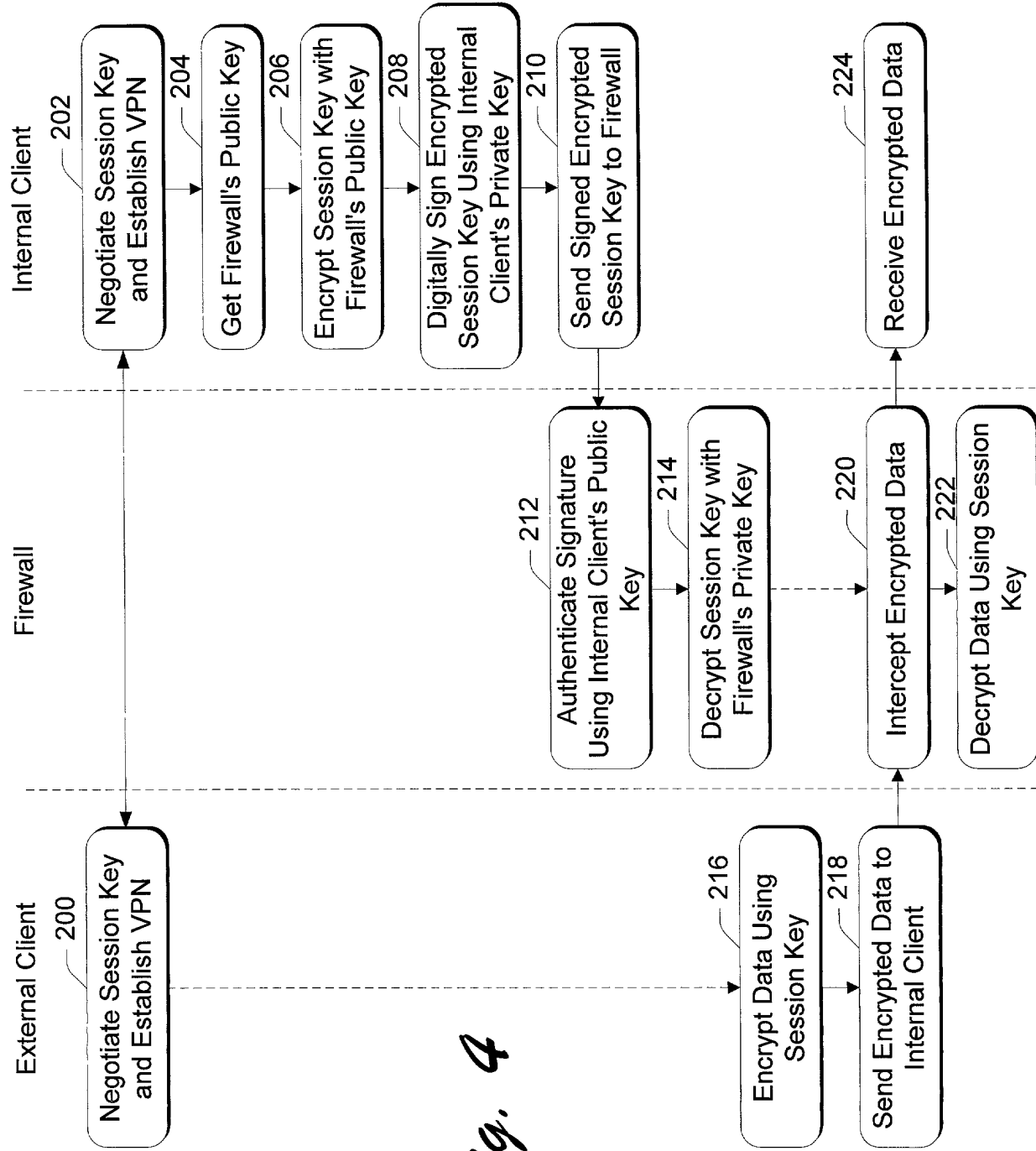


Fig. 4



# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship ..... Gunter et al.  
 Applicant ..... Microsoft Corporation  
 Attorney's Docket No. .... MSI-298US  
 Title: 'System and Method for Trusted Inspection of a Data Stream

## DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled "System and Method for Trusted Inspection of a Data Stream," the specification of which is attached hereto.

I have reviewed and understand the content of the above-identified specification, including the claims.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

PRIOR FOREIGN APPLICATIONS: no applications for foreign patents or inventor's certificates have been filed prior to the date of execution of this declaration.

### **Power of Attorney**

I appoint the following attorneys to prosecute this application and transact all future business in the Patent and Trademark Office connected with this application:  
 Lewis C. Lee, Reg. No. 34,656; Daniel L. Hayes, Reg. No. 34,618; Allan T.

Sponseller, Reg. 38,318; James R. Banowsky, Reg. No. 37,773; Steven R. Sponseller, Reg. No. 39,384; David A. Morasch, Reg. No. 42,905; Katie E. Sako, Reg. No. 32,628 and Daniel D. Crouse, Reg. No. 32,022.

Send correspondence to: LEE & HAYES, PLLC, W. 201 North River Drive, Suite 430, Spokane, Washington, 99201. Direct telephone calls to: Lewis C. Lee (509) 324-9256.

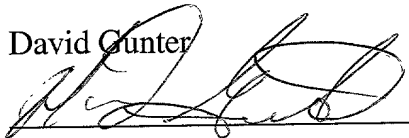
All statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statement may jeopardize the validity of the application or any patent issued therefrom.

\*\*\*\*\*

Full name of inventor:

David Gunter

Inventor's Signature



Date: 3/18/19

Residence:

Redmond, WA

Citizenship:

US

Post Office Address:

17511 NE 22nd Ct.  
Redmond, WA 98052

\*\*\*\*\*

Full name of inventor:

Leeon Moshe Shachaf

Inventor's Signature

Leeon Shachaf

Date: 3/18/99

Residence:

Bellevue, WA

Citizenship:

US/Israel

Post Office Address:

3227 168th Ave. NE  
Bellevue, WA 98008

6600 "The Shachaf"